Pattern Recognition 47 (2014) 3959–3966

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

An irrelevant variability normalization approach to discriminative training of multi-prototype based classifiers and its applications for online handwritten Chinese character recognition

Jun Du [a,*], Qiang Huo [b]

[a] National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP), University of Science and Technology of China, No. 96, JinZhai Road, Hefei, Anhui PR China
[b] Microsoft Research Asia, 13/F, Building 2, No. 5 Danling Street, Haidian District, Beijing, PR China

ARTICLE INFO

Article history:
Received 4 January 2014
Received in revised form
19 May 2014
Accepted 17 June 2014
Available online 24 June 2014

Keywords:
Irrelevant variability normalization
Sample separation margin
Minimum classification error
Rprop
Discriminative training
Online handwritten Chinese character recognition

ABSTRACT

This paper presents an irrelevant variability normalization (IVN) approach to jointly discriminative training of feature transforms and multi-prototype based classifier for recognition of online handwritten Chinese characters. A sample separation margin based minimum classification error criterion is adopted in IVN-based training, while an Rprop algorithm is used for optimizing the objective function. For the IVN approach based on piecewise linear transforms, the corresponding recognizer can be made both compact and efficient by using a two-level fast-match tree whose internal nodes coincide with the labels of feature transforms. Furthermore, the IVN system using weighted sum of linear transforms outperforms that based on piecewise linear transforms. The effectiveness of the proposed approach is first confirmed using an in-house developed online Chinese handwriting corpus with a vocabulary of 9306 characters, and then further verified on a standard benchmark database for an online handwritten character recognition task with a vocabulary of 3755 characters.

1. Introduction

Using online handwritten Chinese character recognition as an input mode on a portable device has been becoming increasingly popular. Good solutions have been developed to build product engines for online handwritten Chinese character recognition (e.g., [1–3]). In spite of many successful applications, however, the problem of *diversified* training data and/or possible *mismatch* between training and testing conditions has not been addressed explicitly in the above solutions. Also the results of the recent Chinese handwriting recognition competition [4,5] reveal the challenge of both isolated character recognition and handwriting text recognition. In this study, we adopt a so-called *irrelevant variability normalization* (IVN) [6] based training strategy to tackle the above problem. IVN is a general concept for pattern recognition problems, which was first proposed in speech recognition area [6]. In a specific pattern recognition problem, there are many irrelevant variabilities in the contents to be recognized, which lead to the degradation of recognition performance in real applications. For example, in speech recognition area, those variabilities could be speaker variability (or accent), background noises, etc. The writing style in online handwriting recognition, the font style, lighting condition, background noises, perspective distortions in the image for optical character recognition (OCR) can be also considered as irrelevant variabilities. The main idea of IVN is to normalize all those variabilities explicitly or implicitly either in the feature space or model space for both training stage and recognition stage.

In [7], a so-called speaker adaptive training (SAT) approach was proposed to normalize speaker variability in training hidden Markov models (HMMs) for automatic speech recognition (ASR). The concept of SAT training was generalized to deal with any variabilities irrelevant to phonetic classification in [6], therefore a term of IVN training was coined, where as an illustrative example, the IVN training was used to improve learning HMM state tying from data based on phonetic decision-tree. Since then, many variants of IVN training methods have been tried in ASR area. For example, IVN-based training of feature transforms and HMMs based on maximum likelihood [8] and discriminative training [9] has been verified to be effective for large vocabulary continuous speech recognition (LVCSR). A region-dependent feature transform (RDT) approach proposed in [10,11], which was a weighted sum of linear transforms using the Gaussian posterior as the weight coefficient, was yet another example of IVN training. Only recently, the concept of IVN training was tried in the area of handwriting recognition. For example, in [12], writer adaptive training (WAT) using constrained maximum likelihood linear regression (CMLLR)

* Corresponding author. Tel.: +86 551 63607863.
E-mail address: jundu@ustc.edu.cn (J. Du).

http://dx.doi.org/10.1016/j.patcog.2014.06.014
0031-3203/

[13] based feature transform was studied for an HMM-based Arabic handwriting recognition task. RDT-based approach in [10] was also applied to HMM-based off-line handwriting recognition in [14]. More recently, a pattern field classification approach with style normalized transformation was proposed in [15] and demonstrated to be effective for several pattern recognition applications, including handwritten Chinese character recognition.

In our recent work [16], we study the problem of IVN-based training for online handwritten Chinese character recognition. One of the state-of-the-art techniques to build a Chinese handwriting recognizer is to use a so-called sample separation margin (SSM) based minimum classification error (MCE) criterion [17,18], which is similar to the generalized learning vector quantization (GLVQ) approach in [35], to train a prototype-based classifier as reported in [1]. In spite of the large vocabulary of Chinese characters, such a classifier can be made both compact (e.g., [19]) and efficient (e.g., [20]) in the recognition stage. In [16], we propose an approach to IVN-based joint training of feature transforms and prototype-based classifier parameters by using the SSM-MCE criterion and demonstrate its effectiveness for Chinese handwriting recognition as an illustrative example. An Rprop algorithm ([21,22]) is used to optimize the objective function. Furthermore, the IVN-trained recognizer can be made both compact and efficient by using a two-level fast-match tree [20] whose internal nodes coincide with the labels of feature transforms. In this paper, we extend the above work in the following ways: (1) the weighted sum of linear transforms, similar to the region dependent linear transforms in [10], is used as the feature transforms which can achieve further improvements of recognition accuracy over the IVN approach based on piecewise linear transforms in [16], (2) the experiments are also conducted on a standard benchmark database beyond the in-house corpus, (3) the detailed descriptions of relevant procedure for Rprop optimization are reported. Our work is also related to the log-likelihood of hypothesis margin (LOGM) approach in [23] and discriminative feature extraction (DFE) work in [24]. The LOGM is a modification of the MCE criterion [25] for improving the training convergence and generalization performance of prototype-based classifier while DFE optimizes a linear feature transform jointly with the classifier parameters. Our experimental results show that our IVN approach performs significantly better than the approach using LOGM and DFE as multiple linear transforms are used.

Fig. 1 illustrates an overall system development flow of our work in this paper. In the first module, after feature extraction of training samples, an LBG clustering algorithm [26] is used to construct multiple prototypes for each character class. Then a baseline classifier is constructed by using the SSM-MCE training. In the second module, the clusters of feature space associated with feature transforms are generated via the baseline classifier, which are used for the IVN-based SSM-MCE joint training of feature transforms and prototype-based classifier parameters. Finally, with the IVN resources from the second module, at recognition stage (i.e., in the third module), the corresponding transform after cluster selection is used to transform the feature vector of the unknown sample, which is then fed to the IVN-based SSM-MCE trained classifier for recognition.

The remainder of the paper is organized as follows. In Section 2, we describe briefly how to construct a multi-prototype based classifier by using the SSM-MCE training. In Section 3, we present the detailed procedure for IVN-based SSM-MCE joint training of feature transforms and classifier parameters. The fast-match technique is introduced in Section 4. In Section 5, we report experimental results of our proposed approach. Finally we conclude the paper in Section 6.

2. SSM-MCE training of a multi-prototype based classifier

Suppose our classifier can recognize M character classes denoted as $\{C_i | i=1,\dots,M\}$. For a multi-prototype based classifier, each class C_i is represented by K_i prototypes, $\boldsymbol{\lambda}_i=\{\mathbf{m}_{ik} \in \mathcal{R}^D | k=1,\dots,K_i\}$, where $\mathbf{m}_{ik}$ is the kth prototype of the ith class. Let's use $\Lambda=\{\boldsymbol{\lambda}_i\}$ to denote the set of prototypes. In the classification stage, a feature vector $\mathbf{y} \in \mathcal{R}^D$ is first extracted. Then $\mathbf{y}$ is compared with each of the M classes by evaluating a Euclidean distance based discriminant function for each class C_i as follows

$$g_i(\mathbf{y};\boldsymbol{\lambda}_i) = -\min_k \|\mathbf{y}-\mathbf{m}_{ik}\|^2. \tag{1}$$

The class with the maximum discriminant function score is chosen as the recognized class $r(\mathbf{y};\boldsymbol{\Lambda})$, i.e.,

$$r(\mathbf{y};\boldsymbol{\Lambda}) = \arg\max_i g_i(\mathbf{y};\boldsymbol{\lambda}_i). \tag{2}$$

In the training stage, given a set of training feature vectors $\mathcal{Y}=\{\mathbf{y}_r \in \mathcal{R}^D | r=1,\dots,R\}$, first we initialize $\boldsymbol{\Lambda}$ by LBG clustering [26]. Then $\boldsymbol{\Lambda}$ can be re-estimated by minimizing the following SSM-MCE objective function:

$$l(\mathcal{Y};\boldsymbol{\Lambda}) = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{1+\exp[-\alpha d(\mathbf{y}_r;\boldsymbol{\Lambda})+\beta]} \tag{3}$$

where α, β are two control parameters, and $d(\mathbf{y}_r;\boldsymbol{\Lambda})$ is a misclassification measure defined by using a so-called sample separation margin (SSM) as follows [17]:

$$d(\mathbf{y}_r;\boldsymbol{\Lambda}) = \frac{-g_p(\mathbf{y}_r;\boldsymbol{\lambda}_p)+g_q(\mathbf{y}_r;\boldsymbol{\lambda}_q)}{2\,\|\mathbf{m}_{p\hat{k}}-\mathbf{m}_{q\bar{k}}\|} \tag{4}$$

where

$$\hat{k} = \arg\min_k \|\mathbf{y}_r-\mathbf{m}_{pk}\|^2 \tag{5}$$

$$q = \arg\max_{i \in \mathcal{M}_r} g_i(\mathbf{y}_r;\boldsymbol{\lambda}_i) \tag{6}$$

$$\bar{k} = \arg\min_k \|\mathbf{y}_r-\mathbf{m}_{qk}\|^2 \tag{7}$$

and $\mathcal{M}_r$ is the hypothesis space for the rth sample, excluding the true label p.

To optimize the objective function in Eq. (3), the same implementation of Rprop algorithm as described in [1] is adopted here.

3. IVN-based SSM-MCE joint training

3.1. Feature transformation

In this study, the concept of IVN is implemented by using feature transformation. Two feature transforms are explored, namely piecewise linear transforms (PLT) and weighted sum of linear transforms (WSLT). For PLT based IVN training, the following feature transformation is used:

$$\mathbf{x}_r = \mathcal{F}_1(\mathbf{y}_r;\boldsymbol{\Theta}) = \mathbf{A}_{e_r}\mathbf{y}_r+\mathbf{b}_{e_r} \tag{8}$$

where $\mathbf{y}_r$ and $\mathbf{x}_r$ are the rth D-dimensional input and transformed feature vectors, respectively; and e_r is the transform label for the rth sample. Let's use $\boldsymbol{\Theta}=\{(\mathbf{A}_e,\mathbf{b}_e)|e=1,\dots,E\}$ to denote the set of transform parameters of E linear transforms, where $\mathbf{A}_e$ is a $D\times D$ nonsingular matrix and $\mathbf{b}_e$ is a D-dimensional bias vector. As for WSLT based IVN training, the corresponding transform is defined as

$$\mathbf{x}_r = \mathcal{F}_2(\mathbf{y}_r;\boldsymbol{\Theta}) = \sum_{e=1}^{E} w_r^e(\mathbf{A}_e\mathbf{y}_r+\mathbf{b}_e) \tag{9}$$

with the constraint

$$\sum_{e=1}^{E} w_r^e = 1, \tag{10}$$

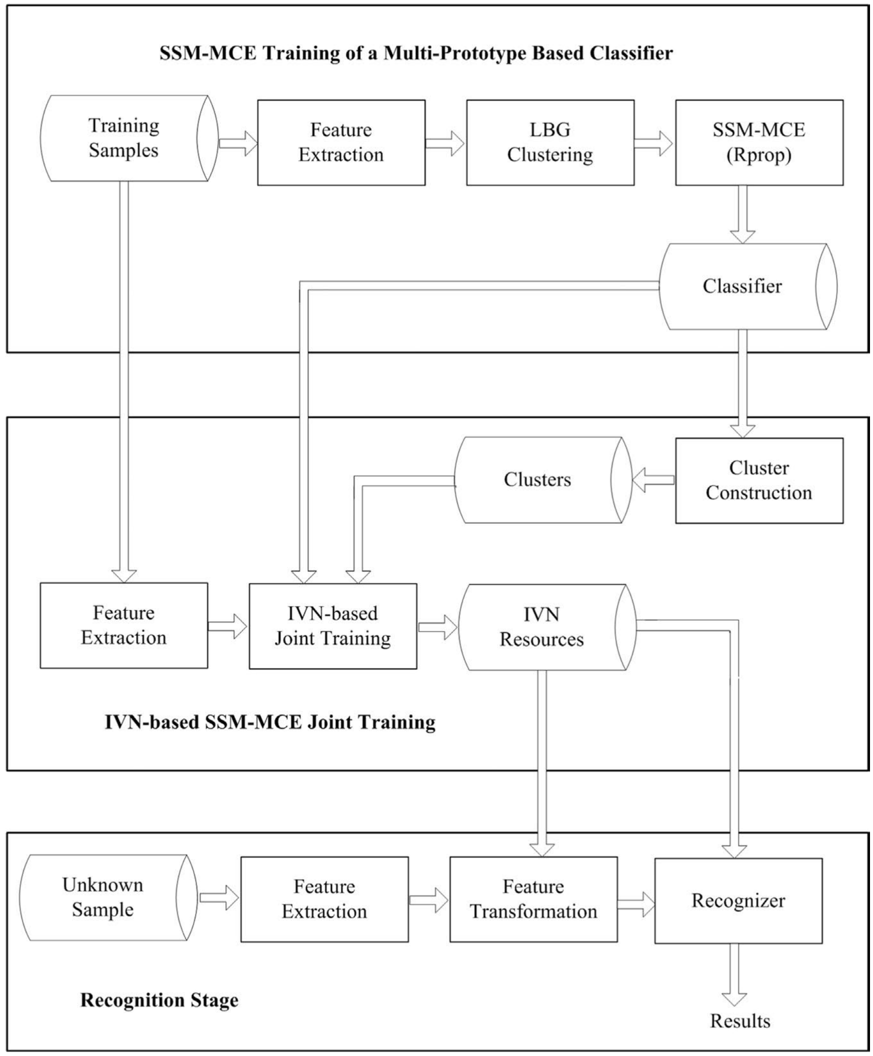

Fig. 1. Overall flow of system development.

where the w_r^e is the weight coefficient of the eth linear transform for the rth sample $\mathbf{y}_r$.

Hopefully the transformed feature vector $\mathbf{x}_r$ in Eq. (8) or Eq. (9) has less irrelevant information to the content to be recognized and finally results in a compact classifier. In the next subsection, we elaborate on how to determine the transform label e_r in Eq. (8) and the weight coefficient w_r^e in Eq. (9).

3.2. Cluster construction and selection

Suppose the feature space can be divided into E clusters and each cluster e is associated with one linear transform $(\mathbf{A}_e, \mathbf{b}_e)$, which is assumed to normalize the irrelevant variability of the feature vector belonging to this cluster. In this work, to construct the clusters, we divide all the prototypes of all classes into E groups by using k-means clustering approach. The centroid of each cluster $\mathbf{c}_e$ is calculated as the sample mean of the prototypes belonging to the cluster e. Then in both IVN-based training and recognition stage based on PLT, given the clusters and for each feature vector $\mathbf{y}_r$, a transform label e_r is assigned to the feature vector as the label of the cluster having the minimum Euclidean distance between the feature vector and the cluster centroid:

$$e_r = \arg\min_e \|\mathbf{y}_r - \mathbf{c}_e\|^2. \tag{11}$$

For WSLT, given the above clusters, the weight coefficient w_r^e in Eq. (9) can be calculated using the softmax function

$$w_r^e = \frac{\exp\{-\tau\|\mathbf{y}_r - \mathbf{c}_e\|^2\}}{\sum_{i=1}^{E} \exp\{-\tau\|\mathbf{y}_r - \mathbf{c}_i\|^2\}}, \tag{12}$$

and τ is estimated from the training set:

$$\tau = \frac{\tau_0 R}{\sum_{r=1}^{R} \min_e \|\mathbf{y}_r - \mathbf{c}_e\|^2}, \tag{13}$$

where τ_0 is a scaling factor. Another way to calculate w_r^e can refer to [10], where the Gaussian posterior probability is used as the weight coefficient for the RDT approach. First a Gaussian mixture model (GMM) with E mixture components can be built on top of the E clusters by using Expectation-Maximization (EM) algorithm:

$$p(\mathbf{y}) = \sum_{e=1}^{E} \omega_e \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e) \tag{14}$$

where $\boldsymbol{\mu}_e$, $\boldsymbol{\Sigma}_e$, and ω_e are the mean vector, diagonal covariance matrix and mixture weight of the eth Gaussian component. Then w_r^e is set as the Gaussian posterior probability:

$$w_r^e = \frac{\omega_e \mathcal{N}(\mathbf{y}_r; \boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e)}{\sum_{i=1}^{E} \omega_i \mathcal{N}(\mathbf{y}_r; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}. \tag{15}$$

3.3. Training procedure

The IVN-based SSM-MCE objective function is defined as follows:

$$l(\mathcal{Y};\mathbf{\Lambda},\mathbf{\Theta})=\frac{1}{R}\sum_{r=1}^{R}\frac{1}{1+\exp[-\alpha d(\mathbf{y}_r;\mathbf{\Lambda},\mathbf{\Theta})+\beta]} \tag{16}$$

where

$$d(\mathbf{y}_r;\mathbf{\Lambda},\mathbf{\Theta})=\frac{-g_p(\mathbf{x}_r;\boldsymbol{\lambda}_p)+g_q(\mathbf{x}_r;\boldsymbol{\lambda}_q)}{2\|\mathbf{m}_{p\hat{k}}-\mathbf{m}_{q\bar{k}}\|}. \tag{17}$$

In the above equations, $\mathbf{x}_r$ is defined in Eq. (8) or Eq. (9), where the corresponding feature transforms can be determined as described in Section 3.2. The following *method of alternating variables* can then be used to jointly estimate $\mathbf{\Theta}$ and $\mathbf{\Lambda}$ by minimizing the above objective function:

Step 1 : *Initialization*
First, the classifier parameters $\mathbf{\Lambda}$ are initialized by using SSM-MCE training described in Section 2. The transform parameters $\mathbf{\Theta}$ are initialized as $\mathbf{b}_e=\mathbf{0}$ and $\mathbf{A}_e=\mathbf{I}$.

Step 2 : *Estimating the feature transform parameters* $\mathbf{\Theta}$ *by fixing the classifier parameters* $\mathbf{\Lambda}$
Given the fixed classifier parameters $\overline{\mathbf{\Lambda}}$, the SSM-MCE objective function $l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta})$ can be optimized by using an Rprop algorithm with N_T iterations as described in Appendix B.

Step 3 : *Estimating the classifier parameters* $\mathbf{\Lambda}$ *by fixing the feature transform parameters* $\mathbf{\Theta}$
Given the updated transform parameters $\overline{\mathbf{\Theta}}$ obtained in Step 2, we first transform each training feature vector $\mathbf{y}_r$ by using Eq. (8) or Eq. (9). Then an Rprop algorithm with N_C iterations is performed as described in Appendix A to re-estimate classifier parameters $\mathbf{\Lambda}$ by minimizing the objective function $l(\mathcal{Y};\mathbf{\Lambda},\overline{\mathbf{\Theta}})$.

Step 4 : *Repeat Step 2 and Step 3* N_{IVN} *times*.
In the above training procedure, the control parameters N_T, N_C, and N_{IVN} are set empirically.

4. Fast match technique

Our fast-match technique is based on a two-level tree [20]. To construct the tree, G clusters are first generated as described in Section 3.2. Each cluster has a bucket consisting of character classes with their prototypes belonging to the cluster. Each training feature vector will then be classified into the cluster with the minimum Euclidean distance between the feature vector and the cluster centroid. The character class of the training sample will be added into the bucket if it is not in the bucket yet. In this way, we obtain a two-level tree with G buckets, each containing a number of character classes. In this work, to make the recognizer both compact and efficient, we share the clusters in PLT based IVN training and fast-match tree, i.e., we set $E=G$. In recognition stage, given the feature vector extracted from an unknown sample, we can find "Top N" candidates efficiently by using the following fast-match procedure:

- Compare the input feature vector with each cluster centroid and sort the result in ascending order of the Euclidean distances, which can be considered as the first-level recognition;
- If PLT based IVN training is performed, the feature transform associated with the first cluster is applied to the input feature vector; Otherwise, skip this step;
- Merge all character classes in the top N_B buckets and use them to perform the second-level recognition as usual.

In the above procedure, a technique known as the *partial distance based elimination* has been used to speed up the process of identifying the "Top N" candidates.

Table 1
The information of the training data for in-house corpus.

Vocabulary	GB2312-L1		GB2312-L2		CJK	Others
Writing style	Regular	Cursive	Regular	Cursive	Regular	Regular
# of characters	3755	3755	3008	3008	2380	163
# of duplicates	0	1	0	0	0	0

5. Experiments and results

5.1. Experimental setup

The experiments are first conducted on an in-house corpus for the task of recognizing isolated online handwritten characters with a vocabulary of 9306 character classes including 9143 Chinese characters, 62 alphanumeric characters, 101 punctuation marks and symbols. As shown in Table 1, this vocabulary mainly consists of 3755 GB2312-L1 characters, 3008 GB2312-L2 characters, 2380 CJK characters, and other 163 characters, which is used in the product engine for Chinese handwriting recognition of Microsoft. For training, we used about 1000 samples averaged per character class. Also from Table 1, the regular-style training samples are collected for all characters while the cursive-style training samples are only covered for GB2312-L1 and GB2312-L2. Because there are much more regular-style samples than cursive ones, even for GB2312-L1 and GB2312-L2, the *re-sampling* of training samples is performed as in [19]. We enlarge the proportion of cursive-style samples by using one more duplicate for the most commonly used GB2312-L1 characters, which is listed in Table 1. Three testing sets are used for evaluation: (1) **Regular-1**: 97,221 samples from 6903 character classes which are written in regular style; (2) **Regular-2**: 84,549 samples from 2355 uncommon character classes in regular style; (3) **Cursive**: 383,064 samples from 3755 frequently used character classes written in cursive style. For feature extraction, a 512-dimensional raw feature vector is extracted as described in [27], which is followed by LDA (linear discriminant analysis) transformation to obtain a lower dimensional feature vector. As for the number of prototypes for each character, we use A prototypes for 3755 most frequently used Chinese characters and B prototypes for the rest of character classes. For Rprop-based SSM-MCE training and IVN-based SSM-MCE joint training, the control parameters are set as described in [1] and [28,29]. Other control parameters are set as: $D=80$, $E=G=128$, $N_T=10$, $N_C=10$, $N_{IVN}=5$, $\tau_0=20$.

To evaluate on a standard benchmark, we also verify our approach on the public database released by the Institute of Automation of Chinese Academy of Sciences (CASIA) [30]. The feature datasets for evaluating isolated online handwritten Chinese character recognition are used which can be downloaded via [31]. The detailed information of the datasets can be found in Table 2. By combining OLHWDB1.0 and OLHWDB1.1 datasets, there are totally 2,154,582 samples in the training set and 538,601 samples in the testing set. The raw feature of the online handwritten character sample is a 512-dimensional vector using the 8-direction histogram of original trajectory direction combined with pseudo 2D bi-moment normalization [32,30]. Then each feature vector is transformed by Box-Cox transformation [33], followed by LDA transformation to obtain a 160-dimensional feature vector. To perform Rprop-based SSM-MCE training and IVN-based SSM-MCE joint training, only the parameter α should be set to 1 due

Table 2
The information of the isolated online handwritten Chinese character database.

Dataset	#Class	Dimension	#Writer (Train)	#Writer (Test)	#Sample (Train)	#Sample (Test)
OLHWDB1.0	3740	512	336	84	1,256,009	314,042
OLHWDB1.1	3755	512	240	60	898,573	224,559

Table 3
Performance (recognition accuracies in %) comparison of baseline systems and IVN-trained systems using piecewise linear transforms on three testing sets under different settings of the number of prototypes and the number of buckets searched in fast-match tree. The footprint (in MB) and runtime latency (normalized by Baseline(2,1) without fast-match) of the corresponding recognizers are also compared.

Method(A,B)	N_B	Regular-1	Regular-2	Cursive	Footprint	Latency
Baseline(2,1)	N/A	96.53	94.84	91.53	4.10	1
Baseline(4,2)	N/A	96.88	94.93	92.28	8.08	1.92
Baseline(8,4)	N/A	97.09	94.66	92.74	16.00	3.66
IVN(2,1)	N/A	96.89	95.32	92.18	7.30	1.06
IVN(4,2)	N/A	97.19	95.54	92.88	11.28	1.99
IVN(8,4)	N/A	97.38	95.32	93.23	19.2	3.75
Baseline(2,1)	5	96.52	94.84	91.53	4.44	0.52
Baseline(4,2)	5	96.88	94.92	92.28	8.41	0.89
Baseline(8,4)	5	97.09	94.65	92.74	16.33	1.58
IVN(2,1)	5	96.89	95.32	92.19	7.60	0.53
IVN(4,2)	5	97.19	95.53	92.88	11.57	0.90
IVN(8,4)	5	97.39	95.31	93.23	19.49	1.59
Baseline(2,1)	3	96.52	94.80	91.52	4.44	0.37
Baseline(4,2)	3	96.88	94.89	92.28	8.41	0.65
Baseline(8,4)	3	97.09	94.62	92.74	16.33	1.17
IVN(2,1)	3	96.89	95.29	92.18	7.60	0.38
IVN(4,2)	3	97.19	95.48	92.87	11.57	0.66
IVN(8,4)	3	97.39	95.27	93.22	19.49	1.18

to the dynamic range of the new feature . All the other parameters are the same as those in the experiments on in-house corpus.

To handle large-scale training data, the tools for LBG clustering, SSM-MCE training, and IVN-based SSM-MCE joint training with the Rprop algorithm have been implemented based upon MSR Asia's MPI-based machine learning platform [34]. This platform was developed on top of Microsoft Windows HPC Server, and optimized for various machine learning algorithms. With this high-performance parallel computing platform, experiments can be run very efficiently for large-scale tasks.

5.2. Experimental results on in-house Corpus

Table 3 summarizes a performance (recognition accuracies in %) comparison of baseline systems and IVN-trained systems using piecewise linear transforms on three testing sets under different settings of the number of prototypes and the number of buckets searched in fast-match tree. The footprint (in MB) and runtime latency (normalized by Baseline(2,1) without fast-match) of the corresponding recognizers are also compared. Here footprint is the size of the resources for the recognition engine including the classifier and the corresponding transformations while the runtime latency refers to the run time of the recognizer averaged on each character sample, which is a normalized version by the system denoted as Baseline(2,1) without fast-match. “Baseline” refers to an SSM-MCE trained system without IVN training while “IVN” refers to a system using our proposed IVN-based joint training. The second column of Table 3 indicates the top N_B buckets selected for second-level recognition in fast-match tree, where “N/A” means no fast-match is used. Three prototype configurations, namely (2,1), (4,2), and (8,4) are listed for comparison, because over-training would be observed if the number of prototypes was increased beyond (8,4) in current experiments. The runtime latency in the last column only includes the recognition time after feature extraction.

Based on those results, several observations can be made. First, IVN systems can achieve consistently significant improvements in recognition accuracy compared with the corresponding Baseline systems on all testing sets. Second, under the same prototype setting, the increased runtime latency from Baseline to IVN systems can be almost ignored, especially in cases with fast-match because the first-level recognition of fast-match tree and the cluster selection for each testing feature vector are shared completely. Third, IVN systems can still outperform Baseline systems with smaller footprints and less runtime latency, e.g., IVN(2,1) *vs.* Baseline(4,2) and IVN(4,2) *vs.* Baseline(8,4). Finally, with fast-match technique, the runtime latency of IVN systems can be reduced significantly while the footprint is only increased slightly. The tradeoff between recognition accuracy and efficiency can be made by setting different N_B. Compared with systems without using fast-match technique, $N_B=5$ keeps the same recognition accuracy with reduced runtime latency while $N_B=3$ degrades slightly recognition accuracy with a much more significant reduction of runtime latency.

Table 4 compares the “Top-N” recognition accuracies of Baseline systems and IVN-trained systems using piecewise linear transforms on three testing sets under different settings of the number of prototypes and a single setting of $N_B=5$ for fast match. From the Top-5 and Top-10 results, IVN systems can achieve very high recognition accuracies already.

5.3. Experimental results on CASIA database

Table 5 shows a performance (recognition accuracies in %) comparison of two prototype-based systems under different settings of the number of prototypes on the training and testing sets. “LBG” denotes a system trained using LBG clustering while “SSM-MCE” refers to a system trained by the SSM-MCE criterion. SSM-MCE systems consistently and significantly outperform LBG systems on both training and testing sets with different number of prototypes. By the nature of SSM-MCE training, the improvements

Table 4
Performance (“Top-N” recognition accuracies in %) comparison of baseline systems and IVN-trained systems using piecewise linear transforms on three testing sets under different settings of the number of prototypes and a single setting of $N_B=5$ for fast match.

Method(A,B)	Top-N	Regular-1	Regular-2	Cursive
Baseline(2,1)	Top-1	96.52	94.84	91.53
	Top-5	99.49	99.45	97.89
	Top-10	99.67	99.73	98.72
Baseline(4,2)	Top-1	96.88	94.92	92.28
	Top-5	99.65	99.52	98.18
	Top-10	99.83	99.75	98.92
Baseline(8,4)	Top-1	97.09	94.65	92.74
	Top-5	99.71	99.53	98.39
	Top-10	99.86	99.80	99.05
IVN(2,1)	Top-1	96.89	95.32	92.19
	Top-5	99.65	99.65	98.20
	Top-10	99.83	99.82	98.94
IVN(4,2)	Top-1	97.19	95.53	92.88
	Top-5	99.73	99.70	98.45
	Top-10	99.87	99.85	99.11
IVN(8,4)	Top-1	97.39	95.31	93.23
	Top-5	99.77	99.73	98.60
	Top-10	99.89	99.87	99.20

Table 5
Performance (recognition accuracies in %) comparison of LBG and SSM-MCE systems under different settings of the number of prototypes on the training and testing sets.

#prototype	LBG		SSM-MCE	
	Train	Test	Train	Test
1	89.81	88.74	95.67	92.48
2	91.26	89.90	97.07	92.89
4	92.64	90.74	98.15	93.03

Table 6
Performance (recognition accuracies in %) comparison of IVN-trained systems using piecewise linear transforms on the training and testing sets under different settings of the number of transforms and the number of prototypes.

#transform	1 Prototype		2 Prototypes		4 Prototypes	
	Train	Test	Train	Test	Train	Test
$E=1$	95.75	92.66	97.23	93.16	98.27	93.33
$E=2$	96.03	92.70	97.29	93.18	98.31	93.37
$E=4$	96.06	92.76	97.39	93.23	98.32	93.40
$E=8$	96.27	92.79	97.41	93.24	98.34	93.40
$E=16$	96.35	92.86	97.50	93.27	98.38	93.41
$E=32$	96.57	92.86	97.73	93.27	98.44	93.44
$E=64$	96.83	92.88	97.81	93.28	98.51	93.44
$E=128$	97.10	**92.90**	97.93	**93.30**	98.59	**93.47**
$E=256$	97.39	92.87	98.10	93.27	98.68	93.43
$E=512$	97.68	92.82	98.28	93.20	98.78	93.39

Table 7
Performance ("Top-N" recognition accuracies in %) comparison of baseline systems and different IVN-trained systems on the testing set under different settings of the number of prototypes.

Method	Top-N	1 Prototype	2 Prototypes	4 Prototypes
Baseline	Top-1	92.48	92.89	93.03
	Top-5	97.82	98.04	98.16
	Top-10	98.58	98.72	98.81
IVN-PLT	Top-1	92.90	93.30	93.47
	Top-5	98.06	98.22	98.33
	Top-10	98.75	98.86	98.93
IVN-WSLT-1	Top-1	**93.18**	**93.45**	**93.59**
	Top-5	**98.19**	**98.31**	**98.38**
	Top-10	**98.85**	**98.92**	**98.98**
IVN-WSLT-2	Top-1	93.00	93.34	93.49
	Top-5	98.11	98.27	98.35
	Top-10	98.78	98.88	98.94

of recognition accuracy over the LBG systems on the training set is more obvious than that on the testing set. To verify that our prototype-based classifier is state-of-the-art, we compare our results with those reported in [30]. The readers can refer to Table 8 in [30]. As our testing set is a combination of OLHWDB1.0 and OLHWDB1.1, the performance of the testing set in Table 5 is the average of "DB1.0" and "DB1.1" datasets in Table 8. With the same feature extraction, first our results of LBG system are comparable to those of the "Cluster" column in Table 8 (e.g., for one prototype case, 88.74% between 89.00% and 87.99%). As for the discriminatively trained classifier, the performance of our SSM-MCE classifier with one prototype, 92.48% is between the "DB1.0" 92.73% and the "DB1.1" 92.15% in the "LOGM+DFE" column of Table 8. But for two and four prototypes, SSM-MCE classifiers achieve higher accuracies than "LOGM+DFE" classifiers. Because DFE is not used in our classifiers, the SSM-MCE criterion with Rprop optimization is obviously superior to LOGM.

Table 6 lists a performance (recognition accuracies in %) comparison of IVN-trained systems using piecewise linear transforms on the training and testing sets under different settings of the number of transforms. For $E=1$, it is similar to the "LOGM+DFE" in [30] where the classifier is discriminatively trained followed by discriminative feature extraction using a global linear transform. With the increased number of transforms, the recognition accuracies on the training set increase monotonically while the number of transforms achieving the best performance on the testing set for different number of prototypes is 128, which is set as default for the following experiments. Further increasing the number of transforms beyond 128 leads to over-training problem.

Table 7 summarizes a performance ("Top-N" recognition accuracies in %) comparison of baseline systems and different IVN-trained systems on the testing set under different settings of the number of prototypes. "Baseline" refers to the SSM-MCE trained system without IVN training. "IVN-PLT" denotes the system using the proposed IVN-based joint training via piecewise linear transforms. "IVN-WSLT-1" and "IVN-WSLT-2" are the IVN-trained systems using weighted sum of linear transforms where the weight is calculated in Eqs. (12) and (15), respectively. First, all the IVN-trained systems can yield significant performance improvements over the corresponding baseline systems on the testing set for different prototype setting. Second, compared with the performance on the in-house corpus in Table 3, IVN-PLT systems achieve similarly relative improvements of recognition accuracy on CASIA database with different feature type and dimension. Third, both IVN-WSLT-1 and IVN-WSLT-2 systems consistently outperform IVN-PLT systems which indicates the "soft" selection of linear transforms in WSLT is more powerful than the "hard" selection of linear transforms in PLT. And IVN-WSLT-1 systems achieve the best performance among all the IVN approaches. Finally, IVN-WSLT-2 systems are inferior to IVN-WSLT-1 systems which may be due to the different distance measures where Euclidean distance is used for our classifier design and weight calculation in IVN-WSLT-1 system while Mahalanobis distance is adopted for weight (Gaussian posterior) calculation in IVN-WSLT-2 system.

6. Conclusion

In this paper, we have proposed an approach to IVN-based SSM-MCE joint training of feature transforms and a prototype-based classifier and demonstrated its effectiveness for online handwritten Chinese character recognition on two large vocabulary tasks as an illustrative example. The IVN-trained recognizer using piecewise linear transforms can be made both compact and efficient by using a two-level fast-match tree whose internal nodes coincide with the labels of feature transforms. Given the consistent improvement of recognition accuracy compared with the corresponding SSM-MCE trained systems without using IVN training, even in the case of smaller footprint and runtime latency, the proposed IVN approach in this study offers a good product solution to construct a handwritten character recognizer to be deployed on mobile devices with limited memory for East Asian languages such as Chinese, Japanese, and Korean.

Conflict of Interest

None declared.

Acknowledgments

The major part of this work was done when Jun Du was an FTE employee of Microsoft Research Asia (MSRA) and a visiting researcher under MSRA's Visiting Young Faculty Program. Jun Du's work was also supported in part by the National Natural Science Foundation of China under Grant no. 61305002.

Appendix A. Rprop optimization procedure for classifier parameters

Step 1 : Let $t=0$. η^+ and η^- $(0<\eta^-<1<\eta^+)$ are the increase factor and decrease factor, respectively. Δ_0 is the initial step-size. $\Delta_{\max}$ and $\Delta_{\min}$ are the upper limit and lower limit of step-size, respectively. Calculate the derivative of $l(\mathcal{Y};\mathbf{\Lambda},\overline{\mathbf{\Theta}})$ w.r.t. each m_{ikd} and update the prototype parameters as follows:

$$m_{ikd}^{(t+1)}=m_{ikd}^{(t)}+\Delta m_{ikd}^{(t)} \tag{18}$$

$$\Delta m_{ikd}^{(t)}\triangleq-\operatorname{sign}\left(\frac{\partial l(\mathcal{Y};\mathbf{\Lambda}^{(t)},\overline{\mathbf{\Theta}})}{\partial m_{ikd}}\right)\Delta_{ikd}^{(t)} \tag{19}$$

where m_{ikd} is the dth element of $\mathbf{m}_{ik}$, $m_{ikd}^{(t)}=m_{ikd}$, $\Delta_{ikd}^{(t)}=\Delta_0$, and

$$\frac{\partial l(\mathcal{Y};\mathbf{\Lambda}^{(t)},\overline{\mathbf{\Theta}})}{\partial m_{ikd}}\triangleq\frac{\partial l(\mathcal{Y};\mathbf{\Lambda},\overline{\mathbf{\Theta}})}{\partial m_{ikd}}\Big|_{\mathbf{\Lambda}=\mathbf{\Lambda}^{(t)}}. \tag{20}$$

Step 2 : Let $t=t+1$. Define

$$S=\frac{\partial l(\mathcal{Y};\mathbf{\Lambda}^{(t-1)},\overline{\mathbf{\Theta}})}{\partial m_{ikd}}\cdot\frac{\partial l(\mathcal{Y};\mathbf{\Lambda}^{(t)},\overline{\mathbf{\Theta}})}{\partial m_{ikd}}. \tag{21}$$

Then, the updating formulas are

$$\Delta_{ikd}^{(t)}=\begin{cases}\min(\eta^+\Delta_{ikd}^{(t-1)},\Delta_{\max}) & \text{if } S>0\\ \max(\eta^-\Delta_{ikd}^{(t-1)},\Delta_{\min}) & \text{if } S<0\\ \Delta_{ikd}^{(t-1)} & \text{else}\end{cases} \tag{22}$$

$$\text{If } S<0,\quad \frac{\partial l(\mathcal{Y};\mathbf{\Lambda}^{(t)},\overline{\mathbf{\Theta}})}{\partial m_{ikd}}=0. \tag{23}$$

$$m_{ikd}^{(t+1)}=m_{ikd}^{(t)}+\Delta m_{ikd}^{(t)}. \tag{24}$$

Step 3 : Repeat Step 2 (N_C-1) times.

In the above procedure, the relevant derivative can be calculated as follows:

$$\frac{\partial l_r}{\partial m_{ikd}}=\alpha l_r(1-l_r)\left[\frac{\delta(i,p)\delta(k,\hat{k})(m_{p\hat{k}d}-x_{rd})-\delta(i,q)\delta(k,\overline{k})(m_{q\overline{k}d}-x_{rd})}{\|\mathbf{m}_{p\hat{k}}-\mathbf{m}_{q\overline{k}}\|}\right.$$
$$\left.-d(\mathbf{y}_r;\mathbf{\Lambda},\overline{\mathbf{\Theta}})\frac{(\delta(i,p)\delta(k,\hat{k})-\delta(i,q)\delta(k,\overline{k}))(m_{p\hat{k}d}-m_{q\overline{k}d})}{\|\mathbf{m}_{p\hat{k}}-\mathbf{m}_{q\overline{k}}\|^2}\right] \tag{25}$$

where

$$l_r=\frac{1}{1+\exp[-\alpha d(\mathbf{y}_r;\mathbf{\Lambda},\overline{\mathbf{\Theta}})+\beta]}, \tag{26}$$

and δ is Kronecker delta function.

Appendix B. Rprop optimization procedure for transform parameters

Step 1 : Let $t=0$. Calculate the derivative of $l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta})$ w.r.t. each A_{edj} and b_{ed}, where A_{edj} is the (d,j)th element of the matrix $\mathbf{A}_e$ and b_{ed} is the dth element of the bias vector $\mathbf{b}_e$. Then update the transform parameters as follows:

$$A_{edj}^{(t+1)}=A_{edj}^{(t)}+\Delta A_{edj}^{(t)} \tag{27}$$

$$\Delta A_{edj}^{(t)}\triangleq-\operatorname{sign}\left(\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial A_{edj}}\right)\Delta_{edj}^{(t)} \tag{28}$$

$$b_{ed}^{(t+1)}=b_{ed}^{(t)}+\Delta b_{ed}^{(t)} \tag{29}$$

$$\Delta b_{ed}^{(t)}\triangleq-\operatorname{sign}\left(\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial b_{ed}}\right)\Delta_{ed}^{(t)} \tag{30}$$

where $b_{ed}^{(t)}=0$, $\Delta_{edj}^{(t)}=\Delta_{ed}^{(t)}=\Delta_0$, and

$$\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial A_{edj}}\triangleq\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta})}{A_{edj}}\Big|_{\mathbf{\Theta}=\mathbf{\Theta}^{(t)}} \tag{31}$$

$$\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial b_{ed}}\triangleq\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta})}{\partial b_{ed}}\Big|_{\mathbf{\Theta}=\mathbf{\Theta}^{(t)}}. \tag{32}$$

Step 2 : Let $t=t+1$. Define

$$S_A=\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t-1)})}{\partial A_{edj}}\cdot\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial A_{edj}} \tag{33}$$

$$S_b=\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t-1)})}{\partial b_{ed}}\cdot\frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial b_{ed}} \tag{34}$$

Then, the updating formulas are

$$\Delta_{edj}^{(t)}=\begin{cases}\min(\eta^+\Delta_{edj}^{(t-1)},\Delta_{\max}) & \text{if } S_A>0\\ \max(\eta^-\Delta_{edj}^{(t-1)},\Delta_{\min}) & \text{if } S_A<0\\ \Delta_{edj}^{(t-1)} & \text{else}\end{cases} \tag{35}$$

$$\Delta_{ed}^{(t)}=\begin{cases}\min(\eta^+\Delta_{ed}^{(t-1)},\Delta_{\max}) & \text{if } S_b>0\\ \max(\eta^-\Delta_{ed}^{(t-1)},\Delta_{\min}) & \text{if } S_b<0\\ \Delta_{ed}^{(t-1)} & \text{else}\end{cases} \tag{36}$$

$$\text{If } S_A<0,\quad \frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial A_{edj}}=0 \tag{37}$$

$$\text{If } S_b<0,\quad \frac{\partial l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta}^{(t)})}{\partial b_{ed}}=0 \tag{38}$$

$$A_{edj}^{(t+1)}=A_{edj}^{(t)}+\Delta A_{edj}^{(t)} \tag{39}$$

$$b_{ed}^{(t+1)}=b_{ed}^{(t)}+\Delta b_{ed}^{(t)}. \tag{40}$$

Step 3 : Repeat Step 2 (N_T-1) times.

In the above procedure, the relevant derivatives can be calculated. For the case of piecewise linear transforms, the formulations are

$$\frac{\partial l_r}{\partial A_{edj}}=\frac{\alpha l_r(1-l_r)(m_{q\overline{k}d}-m_{p\hat{k}d})\delta(e,e_r)y_{rj}}{\|\mathbf{m}_{p\hat{k}}-\mathbf{m}_{q\overline{k}}\|} \tag{41}$$

$$\frac{\partial l_r}{\partial b_{ed}} = \frac{\alpha l_r(1-l_r)(m_{q\bar{k}d} - m_{p\hat{k}d})\delta(e, e_r)}{\| \mathbf{m}_{p\hat{k}} - \mathbf{m}_{q\bar{k}} \|} \tag{42}$$

while for the case of weighted sum of linear transforms, the corresponding derivatives are

$$\frac{\partial l_r}{\partial A_{edj}} = \frac{\alpha l_r(1-l_r)(m_{q\bar{k}d} - m_{p\hat{k}d})w_r^e y_{rj}}{\| \mathbf{m}_{p\hat{k}} - \mathbf{m}_{q\bar{k}} \|} \tag{43}$$

$$\frac{\partial l_r}{\partial b_{ed}} = \frac{\alpha l_r(1-l_r)(m_{q\bar{k}d} - m_{p\hat{k}d})w_r^e}{\| \mathbf{m}_{p\hat{k}} - \mathbf{m}_{q\bar{k}} \|} \tag{44}$$

where

$$l_r = \frac{1}{1+\exp[-\alpha d(\mathbf{y}_r; \overline{\mathbf{\Lambda}}, \mathbf{\Theta}) + \beta]}. \tag{45}$$

References

[1] J. Du, Q. Huo, Designing compact classifiers for rotation-free recognition of large vocabulary online handwritten Chinese characters, in: Proc. ICASSP-2012, 2012, pp. 1721–1724.
[2] Y.-Q. Wang, Q. Huo, Building compact recognizers of handwritten Chinese characters using precision constrained Gaussian model, minimum classification error training and parameter compression, Int. J. Doc. Anal. Recognit. 14 (3) (2011) 255–262.
[3] C.-L. Liu, S. Jaeger, M. Nakagawa, Online recognition of Chinese characters: the state-of-the-art, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2) (2004) 198–213.
[4] C.-L. Liu, F. Yin, Q.-F. Wang, D.-H. Wang, ICDAR 2011 Chinese handwriting recognition competition, in: Proc. ICDAR-2011, 2011, pp. 1464–1469.
[5] F. Yin, Q.-F. Wang, X.-Y. Zhang, C.-L. Liu, ICDAR 2013 Chinese handwriting recognition competition, in: Proc. ICDAR-2013, 2013, pp. 1464–1470.
[6] Q. Huo, B. Ma, Irrelevant variability normalization in learning HMM state tying from data based on phonetic decision-tree, in: Proc. ICASSP-1999, 1999, pp. 577–580.
[7] T. Anastasakos, J. McDonough, R. Schwartz, J. Makhoul, A compact model for speaker-adaptive training, in: Proc. ICSLP-1996, 1996, pp. 1137–1140.
[8] G.-C. Shi, Y. Shi, Q. Huo, A study of irrelevant variability normalization based training and unsupervised online adaptation for LVCSR, in: Proc. Interspeech-2010, 2010, pp. 1357–1360.
[9] Y. Zhang, J. Xu, Z.-J. Yan, Q. Huo, A study of an irrelevant variability normalization based discriminative training approach for LVCSR, in: Proc. ICASSP-2011, 2011, pp. 5308–5311.
[10] B. Zhang, S. Matasoukas, R. Schwartz, Discriminatively trained region dependent feature transforms for speech recognition, in: Proc. ICASSP-2006, 2006, pp. 1520–1523.
[11] B. Zhang, S. Matasoukas, R. Schwartz, Recent progress on the discriminative region-dependent transform for speech feature extraction, in: Proc. INTERSPEECH-2006, 2006.
[12] P. Dreuw, D. Rybach, C. Gollan, H. Ney, Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition, in: Proc. ICDAR-2009, 2009, pp. 21–25.
[13] M.J.F. Gales, Maximum likelihood linear transformations for HMM-based speech recognition, Comput. Speech Lang. 12 (2) (1998) 75–98.
[14] J. Chen, B. Zhang, H. Cao, R. Prasad, P. Natarajan, Applying discriminatively optimized feature transform for HMM-based off-line handwriting recognition, in: Proc. ICFHR-2012, 2012, pp. 219–224.
[15] X.-Y. Zhang, K. Huang, C.-L. Liu, Pattern field classification with style normalized transformation, in: Proc. IJCAI-2011, 2011, pp. 1621–1626.
[16] J. Du, Q. Huo, An irrelevant variability normalization based discriminative training approach for online handwritten Chinese character recognition, Proc. ICDAR-2013, 2013, pp. 69–73.
[17] T. He, Q. Huo, A study of a new misclassification measure for minimum classification error training of prototype-based pattern classifiers, in: Proc. ICPR-2008, 2008.
[18] Y.-Q. Wang, Q. Huo, Sample-separation-margin based minimum classification error training of pattern classifiers with quadratic discriminant functions, in: Proc. ICASSP-2010, pp. 1866–1869.
[19] Y.-Q. Wang, Q. Huo, A study of designing compact recognizers of handwritten Chinese characters using multiple-prototype based classifiers, in: Proc. ICPR-2010, 2010, pp. 1872–1875.
[20] Z.-D. Feng, Q. Huo, Confidence guided progressive search and fast match techniques for high performance Chinese/English OCR, in: Proc. ICPR-2002, pp. III-89–92.
[21] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the Rprop algorithm, in: Proc. International Conference on Neural Networks, 1993, pp. 586–591.
[22] C. Igel, M. Hüsken, Improving the Rprop learning algorithm, Proceedings of International Symposium on Neural Computation, 2000, pp. 115–121.
[23] X.-B. Jin, C.-L. Liu, X. Hou, Regularized margin-based conditional log-likelihood loss for prototype learning, Pattern Recognit. 43 (7) (2010) 2428–2438.
[24] C.-L. Liu, R. Mine, M. Koga, Building compact classifier for large character set recognition using discriminative feature extraction, in: Proc. ICDAR-2005, 2005, pp. 846–850.
[25] B.-H. Juang, W. Chou, C.-H. Lee, Minimum classification error rate methods for speech recognition, IEEE Trans. Speech Audio Process. 5 (3) (1997) 257–265.
[26] Y. Linde, A. Buzo, R. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. 28 (1) (1980) 84–95.
[27] Z.-L. Bai, Q. Huo, A study on the use of 8-directional features for online handwritten Chinese character recognition, in: Proc. ICDAR-2005, 2005, pp. 262–266.
[28] J. Du, Q. Huo, A discriminative linear regression approach to OCR adaptation, in: Proc. ICPR-2012, 2012, pp. 629–632.
[29] J. Du, Q. Huo, A discriminative linear regression approach to adaptation of multi-prototype based classifiers and its applications for Chinese OCR, Pattern Recognit. 46 (8) (2013) 2313–2322.
[30] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, Online and offline handwritten Chinese character recognition: benchmarking on new databases, Pattern Recognit. 46 (1) (2013) 155–162.
[31] ⟨http://www.nlpr.ia.ac.cn/databases/handwriting/Download.html⟩.
[32] C.-L. Liu, K. Marukawa, Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition, Pattern Recognit. 38 (12) (2005) 2242–2255.
[33] R.V.D. Heiden, F.C.A. Gren, The Box-Cox metric for nearest neighbor classification improvement, Pattern Recognit. 30 (2) (1997) 273–279.
[34] Z.-J. Yan, T. Gao, Q. Huo, Designing an MPI-based parallel and distributed machine learning platform on large-scale HPC clusters, in: 2012 International Workshop on Statistical Machine Learning for Speech Processing, Kyoto, Japan, March 31, 2012 (⟨http://www.ism.ac.jp/IWSML2012/⟩).
[35] A. Sato, K. Yamada, A formulation of learning vector quantization using a new misclassification measure, in: Proc. ICPR-2008, 2008, pp. 322–325.

Jun Du received the B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively. From 2004 to 2009, he was with the iFlytek Speech Lab of USTC, where he conducted research on speech recognition. During the above period, he worked as an Intern twice for 9 months at Microsoft Research Asia (MSRA), Beijing, China, doing research on discriminative training and noise- robust front-end for speech recognition, and speech enhancement. In 2007, he also worked as a Research Assistant for 6 months at the Department of Computer Science, The University of Hong Kong, doing research on robust speech recognition. From July 2009 to June 2010, he worked at iFlytek Research on speech recognition. From July 2010 to January 2013, he joined MSRA as an Associate Researcher, working on handwriting recognition, OCR, and speech recognition. Since February 2013, Dr. Du worked at National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP) of USTC.

Qiang Huo (M'95) is a Senior Researcher and Research Manager in Microsoft Research Asia (MSRA), Beijing, China. Prior to joining MSRA on August 1, 2007, he had been a faculty member at the Department of Computer Science, The University of Hong Kong since 1998, where he also did his Ph.D. research on speech recognition during 1991 to 1994. From 1995 to 1997, Dr. Huo worked at the ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan. In the past 25 years, he has been doing research and making contributions in the areas of speech recognition, handwriting recognition, OCR, gesture recognition, biometric-based user authentication, hardware design for speech and image processing. Dr. Huo received the B.Eng. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1987, the M.Eng. degree from Zhejiang University, Hangzhou, China, in 1989, and the Ph.D. degree from the USTC in 1994, all in electrical engineering.